논문 2005-42TC-12-19

FPGA를 이용한 Single Rate Three Color Marker 설계

(Design of Single Rate Three Color Marker using FPGA)

박 천 관*

(ChunKwan Park)

요 약

본 논문은 RFC2697에서 제안된 srTCM (single rate Three Color Marker)의 설계에 관한 것이다. srTCM은 AF PHB(Assured Forwarding Per Hop Behavior)를 제공하는 DS (Differentiated Service) 네트워크의 Ingress에서 사용되도록 제안되었다. srTCM은 metering 기능을 가진 marker이며, 토큰 갱신기능과 marking 기능으로 구성된다. 토큰 갱신 기능은 듀얼 토큰버킷을 사용하며, marking 기능은 토큰 값을 입력 패킷의 길이와 비교하여, 그 결과를IP QoS 필드 (ToS 필드)에 마킹한다. 본 논문에서는 이와 같은srTCM를 VHDL과 FPGA를 이용하여 설계하여 하나의 chip으로 구현하였다.

Abstract

This paper addresses the design of srTCM (single rate Three Color Marker) described in RFC2697. It has proposed for srTCM to be used at the ingress of Differentiated Service network that provides AF PHB (Assured Forwarding Per Hop Behavior). srTCM is the marker with the metering function, and consists of the token updating function and the marking function. The token updating function uses the dual token bucket. The marking function compares the token values with the length of input packet, and then marks the result into IP QoS field (ToS field). This paper designs srTCM function and then implements it in one chip using VHDL and FPGA technology.

Keywords: Differentiated, IntServ, DiffSerV, srTCM, PHB, DSCP

I. 서 론

현재 인터넷에서는 모든 패킷이 동일하게 처리되는 최선형 (Best Effort) 서비스만 제공하기 때문에, 서비스에 따른 지연과 지연 변이에 대한 요구사항을 보장해 줄 수 없다. 그러므로 인터넷상에서 QoS을 보장해 주기 위해서 최선형 서비스 모델과 다른 새로운 서비스 모델이 필요하다[1][2]. 이런 추세에 따라 IETF는 인터넷상에서 IP QoS를 보장해 주기 위하여 각 플로우에 대한 상태 정보를 이용하는 통합형 서비스 모델 (IntServ model)과 PHB (Per Hop Behavior)를 이용하는 차별화된 서비스 모델 (DiffServ model)을 제안하였다. IntServ model은 각각의 IP 흐름에 대한 상태 정보를 이용하기 때문에, 트래픽 특성에 따라 QoS를 만족시켜 줄 수 있다. 그러나 이 모델에서는 플로우 수가 증가함에 따라 플로우의 상태 정보량이 증가하기 때문에 네트워크 요소 (라우터 등)들은 많은 저장공간이 필요하다. 이런 이유 때문에 IntServ model은 인터넷과 같은 대규모 공중망에 적용하기 어렵다. DiffServ model은 지연 및 손실 민감도에 따라 서로 다른 트래픽에 차별화된 서비스를 제공하기 위하여 잘 알려진서비스 클래스를 이용한다. DS (Differentiated Service) 네트워크에서, 네트워크 요소들은 각 플로우 대신 플로우들이 속한 aggregate 트래픽 스트림에 따라 트래픽을 서로 다르게 취급한다. 그러므로 DiffServ model은 각 플로우에 대한 상태 및 신호 정보가 필요 없기 때문에 인터넷상에 다양한 서비스를 제공할 수 있다. 따라서, DiffServ model은 인터넷과 같은 대규모 네트워크에 적용할 수 있다.

DS 네트워크에서, 네트워크 자원을 효율적으로 이용하고 모든 사용자에게 다양한 QoS를 제공하기 위하여,

* 정회원, 국립목포해양대학교
(Mokpo National Maritime University)
접수일자: 2005년10월19일, 수정완료일: 2005년12월9일

트래픽 조절 기능이 네트워크 입구에 설치될 필요가 있다. DS 노드는 분류기와 조절기로 구성된다. 패킷이 분류기에 도착하면, 패킷을 서비스에 따라 분류하여 트래픽 조절기로 forwarding 한다. 트래픽 조절기는 미터, 마커, shaper, 그리고 dropper과 같은 4가지 요소로 구성된다. 그러나 상황에 따라 4가지 요소 모두가질 필요는 없다. RFC2696에서, DS네트워트에 대한 트래픽 조절기인 srTCM을 제안하였으며, 이는 마커와 미터로 구성된다[3][4].

srTCM[4]에서, 미터는 TCA (Traffic Conditioning Agreement)에 명시된 트래픽 프로필에 따라 선택된 패킷의 순간적인 특성을 측정하여 측정된 결과를 마커에게 넘겨준다. 마커는 측정 결과에 따라 패킷을 green, yellow, 그리고 red로 마크해 준다. 그것은 패킷의 DS 필드를 특정 codepoint 값으로 세팅함으로써 선택된 패킷의 프로필 준수 여부를 마크하는 것이다. 따라서 본 논문에서는 RFC2697에서 제안한 srTCM를 VHDL 및 FPGA 기술을 이용하여 하나의 chip으로 설계하여 구현하였다.

본 논문 구성은 Ⅱ장에서 DS 네트워크에 대하여 언급하고, Ⅲ장에서는 srTCM의 구조와 동작 알고리즘을 언급한다. 그리고 Ⅳ장에서는 VHDL과 FPGA 기술을 이용하여 srTCM을 하나의 chip으로 설계하고 구현한 것을 언급하고, Ⅴ장에서 결론을 맺는다.

Ⅱ. DS (Differentiated Service) 네트워크

그림 1은 DS (Differentiated Service) 네트워크의 기본구조와 구성요소를 나타낸다. 이 그림에서 DS망 사용자는 먼저 DS망 관리자와 서비스 사용을 위한 협의를 한다. 이것은 쌍방간 합의된 SLA (Service Level Agreement) 이며, 사용자는 SLA에 의하여 DS망을 통하여 전달하려는 패킷 플로우의 집합을 정의하게 된다. DS망의 경계 라우터는 SLA에 의하여 정의된 패킷 플로우의 집합체에 대한 트래픽 분류와 조절기능을 수행한다. 패킷 조절기능은 트래픽 분류에 따른 표시(Mark), 플로우의 측정 (Meter), 쉐이핑(Shaping)과 폴리싱 (Policing) 기능이 있다. DS 망 내부에서는 경계에서 표시된 코드에 의하여 단순히 패킷을 전달하게 된다. 이와 같이 DS 망 내에서 패킷전달기능을 PHB (Per-Hop Behavior)라 한다[2][3].

따라서 DiffServ 모델은 ToS (Type of Service)를 DS (Differentiated Service) 필드로 다시 정의하여

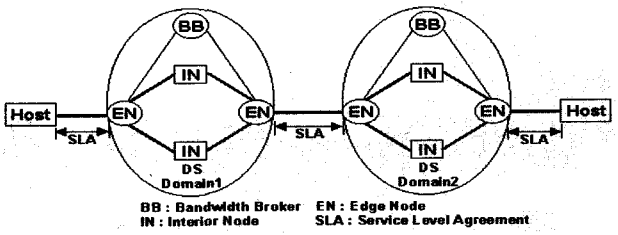


그림 1. DS 네트워크 구성
Fig. 1. DS Network Configuration.

PHP라 하는 기본적인 패킷 전송방법을 정의하고, 이에 따라 패킷의 DS 필드를 어플리케이션에 따라 다르게 표시하고, 이 표시에 따라 패킷을 처리함으로써 몇 개의 차별화된 서비스 클래스를 생성하는 것으로 상대적인 우선순위 기법을 사용하는 모델이다.

DiffServ 모델은 패킷 분류와 같은 기능들을 망의 가장자리에서만 발생하게 하고 망 내부에서는 간단한 패킷 전달기능만 수행하도록 하였다. 따라서 망의 경계 라우터에서는 여러 플로우가 집합된 서비스에 따라 패킷을 분류하고 분류된 패킷에 표시를 한다. 망 내부의 라우터는 패킷에 표시된 정보에 따라 단순히 패킷 전달기능만 수행한다. 또 자원예약 관점에서는 DiffServ 모델에서는 망의 경계 라우터에서만 수행되고, 사용자와 서비스 계약에 따라 고정적으로 이루어질 수 있다. 사용자의 계약 준수 여부도 망의 경계노드에서만 감시된다.

한 도메인에서 DSCP (Differentiated Service Code Point)의 변환 방법은 동일하기 때문에, 균일한 서비스를 제공하는 것이 가능하다. DS 도메인에서 라우터는 엣지 노드 또는내부 노드일 수 있다. 우선, 사용자는 서비스 사용에 대하여 서비스 매니저와 SLA을 통하여 협의한다. SLA는 사용자가 전송할 총계적인 패킷 플로우에 대하여 정의해 놓은 것이다. 엣지 라우터는 SLA에 의하여 패킷을 분류한 후 제어한다. DS 네트워크 내부의 요소들은 패킷을 엣지에서 표시된 코드에 의해서만 전달한다. 이와 같은 패킷 전송 기능을 PHB (Per-Hop Behavior)라 한다[4]. PHB는 모든 라우터에서 유효하며, PHB 부분만 중간 라우터에서 구현된다. 그러므로 중간 라우터는 DS 서비스에 대하여 최소의 기능과 오버헤드를 갖고 대부분의 복잡한 기능은 에지 노드에서 구현된다.

그림 2는 DS 네트워크에서 트래픽 conditioner를 보여준다. 트래픽 conditioner에는 마크 기능, 미터 기능, 그리고shaping/dropper 기능이 있다. 패킷이 분류기에 도착하면, 분류기는 패킷을 분류해서 트래픽

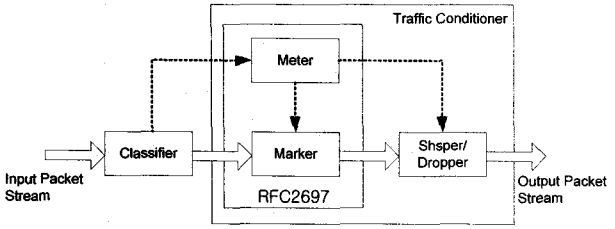


그림 2. DS 망에서 Traffic Conditioner
Fig. 2. Traffic Conditioner in DS Network.

conditioner로 전송한다. 트래픽 conditioner는 DS 노드와 서비스의 요구 조건에 따라 4개 요소 모두 구비할 필요는 없다. RFC2697은마커와 미터를 갖는 트래픽 conditioner을 제안하였다.

인터넷상에서 QoS를 제공하기 위한 Diffserv 모델은 각 네트워크 노드에서 특정한 forwarding 처리 또는 PHB를 받기 위하여 패킷의 표준화된 DS 바이트를 마킹하는 것이다. 패킷은 IPv4 인경우 헤더의 ToS 필드, 또는 IPv6인 경우 트래픽 클래스 필드의 DS 옥텟에 의하여 분류된다. 그림 3은 ToS 바이트와 DS 바이트간 관계를 보여준다. DSCP는 IP precedence에 의하여 사용된 3 비트로의 확장이다. IP precedence와 같이, DSCP는 적당히 표시된 패킷에 서로 다른 처리를 제공하기위하여 사용될 수 있다. DSCP 필드는 IP precedence와 유사하게 IP 헤더의 일부분이다. 사실 DSCP 필드는 IP precedence 필드의 일부분이다. 그러므로 DSCP 필드는 IP precedence 대하여 언급된 것과 유사한 방법으로 사용되고 세트되며, DSCP 필드의 정의는 IP precedence 값과 backward 호환성을 갖는다[5].

Diff-Serv에서 포워딩 결정은 DS 바이트에서 정의된 매개변수에 따라 이루어지며, PHB로 나타난다. 예를 들면, Diff-Serv 네트워크에서 디폴트 PHB는 FIFO(First-In First-Out) 큐잉을 이용하는 기존의 최선형 서비스(Best Effort Service) 이며, 좀 더 높은 서비스 등급을 위하여 다른 PHB가 정의된다. 한가지 예는 신속한 포워딩(EF : Expedited Forwarding)이다. EF가 Diff-Serv 라우터로 입력되면, 그들은 짧은 큐에서 처리되고, 낮은 지연시간, 패킷손실, 그리고 지터를 유지하도록 빨리 서비스되는 것을 의미한다. 가변적인 우선순위를 허용하지만, 패킷이 정확한 순서로 도착하는 것을 보장해 주는 다른 PHB는 AF (Assured Forwarding)가 있다. Diff-Serv는 PHB 분류, 마킹, 그리고 순서 절차를 표준화하였지만, 장비 제조업체와 서비스 제공업자가 PHB를 자유스럽게 할당할 수 있으며, 라우터 밴더는 효율적인 QoS 제어를 구비해 줄 수 있

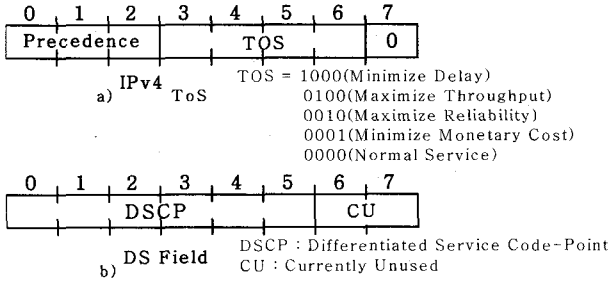


그림 3. ToS 및 DSCP 필드
Fig. 3. ToS and DSCP Field.

을 것으로 생각하는 매개변수와 능력을 정의할 것이며, 서비스 제공업자는 자신들의 QoS를 차별화하기 위하여 PHB 조합을 고안할 수 있다.

DS 네트워크에서, 기존의 BE(Best Effort) 서비스를 제외하고, 두가지 다른 종류의 서비스, 즉 AF(Assured Forwarding) 서비스와 EF (Expedited Forwarding) 서비스가 있다. IETF는 IP 헤더의 특정 바이트를 이용하여 BE 서비스와 두 서비스가 차이점을 언급했다[5][6].

EF PHB는 네트워크 제어관련 트래픽(라우팅 정보의 갱신) 과 같이, 최우선 등급의 전송 방법을 나타낸다. EF PHB을 사용하여 패킷을 전송하는 경우, 이 그룹의 출발 속도는 출발 속도와 같던지 또는 단지버퍼에서 지연 시간 정도 더 크다. DS 네트워크에서, 패킷이 일단 EF 서비스로 마킹되면 DS 도메인내에서 EF 서비스로 forwarding 되던지 또는 삭제된다.

마킹은패킷의 DS 필드를 특정 codepoint로 세팅함으로써 미터링 결과를 반영하는 것이다. AF PHB의 경우, 칼라는 패킷의 삭제 우선권으로 코드화될 수 있다. RFC2597은 AF codepoint 값을 <표 1>에서와 같이 권고하고 있다. 그것은 4개의 AF 트래픽 클래스를 정의하고 있다. 각 서비스 클래스는 3개의 drop precedence 레벨을 가지고 있으며, 각각의 AF 트래픽 클래스는 그 지신의 큐를 가진 서비스로, 4개의 트래픽 클래스에 대하여 WFQ (Weighted Fair Queuing)와 같은 독립적인 용량 관리를 가능하게 한다. 각각의 AF 클래스내에, RED (Random Early Detection)와 같은 큐 관리 기능

표 1. AF Codepoint 및 Color Code[6][7]
Table 1. AF Codepoint and Color Code.

Dropping 우선권	Class 1	Class 2	Class 3	Class 4	Color
Low	001010	010010	011010	100010	Green
Medium	001100	010100	011100	100100	Yellow
High	001110	010110	011110	100110	Red

을 갖는 3개의 drop precedence 레벨이 있다. AF codepoint는 총 6비트로 구성된다. 6비트중 상위 3비트는 해당 패킷의 클래스를 나타내며 나머지 하위 3비트는 해당 클래스내에서 drop precedence 레벨을 나타낸다.

III. srTCM 구조 및 알고리즘

srTCM은 IP 패킷 스트림을 측정한 후 해당 패킷을 green, yellow, 또는 red로 표시한다. srTCM에서 마킹은 3가지 파라미터인, CIR (Committed Information Rate), CBS (Committed Burst Size), 그리고 EBS (Excess Burst Size)이 근간을 두고 있다. 그림 4는 RFC2697에서 제안된 srTCM 구조이며, 미터와 마커로 구성된다[3][4].

미터는두개의 토큰 버킷 c와 e로 구성된다. 미터의 기능은 각 패킷을측정한 후 패킷과 측정 결과를 마커에게 전달해 준다. 마커의 기능은 미터에서 전달된 측정 결과에 따라 패킷의 DS 필드를 특정 codepoint로 셋팅 시켜 주는 것이다.

그림 5는 srTCM에서 미터의 토큰 버킷 갱신 알고리즘을 나타낸다. 이 알고리즘은 두개의 토큰 버킷 c와 e의 항으로 서술된다. 두개의 토큰 버킷은 공통의 토큰 발생 속도 CIR를 공유한다.

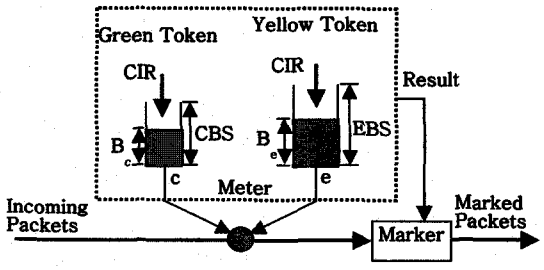


그림 4. srTCM 구조
Fig. 4. SrTCM Architecture.

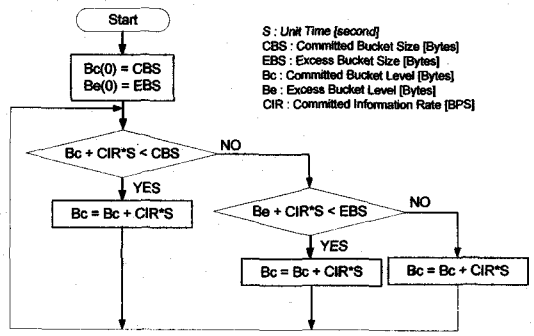


그림 5. 토큰 버킷 갱신 알고리즘
Fig. 5. Updating Algorithm of Token Bucket.

Color-Blind Mode	Color-Award Mode

If L < =Bc
then
acket is marked reen
Bc = Bc - L;
else
if L < =Be
then
acket is marked ellow
Be <= Be - L;
else
acket is marked red
end if;
end if;
	If ((L < =Bc) and (color = reen))
	then
	acket is marked reen
	Bc = Bc - L;
	else
	if ((L < =Be) and (color= reen or ellow))
	then
	acket is marked ellow
	Be <= Be - L;
	else
	acket is marked red
	end if;
	end if;

그림 6. srTCM에서 Metering 알고리즘
Fig. 6. Metering Algorithm in srTCM.

두 개의 버스트 크기 CBS와 EBS는 바이트 단위로 측정된다. 각각의 버스트 사이즈는 각각의 토큰 버킷 사이즈와 관련된다. 토큰 버킷 c의 최대 크기는 CBS이며, 토큰 버킷 e의 최대 크기는 EBS이다. 초기에 두개의 버킷 레벨, Bc와 Be는 각각 Bc(0)=CBS와 Be(0)=EBS로 설정된다.

그림 6은 srTCM의 미터링 알고리즘을 보여준다. 이 알고리즘은 두개의 서로 다른 모드로 이루어지며 초기에 세트된다. Color-blind 모드에서, 미터는 입력 패킷이 어떠한 칼라값도 가지지 않는다고 가정한다. 따라서 이모드에서는 모든 패킷은 동일한 방법으로 처리된다. Color-aware 모드에서는, 미터로 입력되는 패킷 스트림이 이전 노드에서 칼라 값 (Green, Yellow, 또는 Red)을 부여 받은 경우를 말한다. 이 모드에서는 패킷이 부가적으로 칼라 값에 따라서로 다른 방법으로 처리된다. 그림 6에서, L은 입력 패킷의 길이이며 바이트 단위로 측정된다. 그리고 Bc와 Be는 토큰 버킷의 레벨이며 또한 바이트로 측정된다. 마킹은 CIR 및 두개의 각 버킷의 버스트 사이즈, CBS와 EBS에 의하여이루어진다. 입력 패킷의 길이가 CBS를 초과하지 않으면 그패킷은 "Green"으로 마킹되고, 입력 패킷의 길이기 CBS를 초과하고 EBS를 초과하지 않으면, 그패킷은 "Yellow"로 마킹된다. 그리고 위조건 두가지를 모두 만족하지 못하면 "Red"로 마킹된다. 이런 기능을 가진 srTCM은 ingress에서 서비스에 대한 폴리싱 기능을 수행한다. 그와 같은 폴리싱 매커니즘에서, 버스트의 피크 값이아닌 버스트의 길이만 서비스의 적격성을 결정해 줄 수 있다.

IV. srTCM의 구현

그림 7은 III장에서 언급한 srTCM 구조에 따른 블럭도를 보여준다. srTCM chip은 CPU 인터페이스, 두개의 MII (Medium Independent Interface) 인터페이스,

디코더 블록, 버킷갱신 블럭, 미터링 블록, 토큰갱신 블럭, 그리고 path 블록으로 구성된다. CPU 인터페이스는 srTCM의 초기값 (CIR, CBS, EBS, unit time, 그리고 클럭 속도)을 할당해 준다. 이 chip은 또한 이더넷 스위칭과 같은 표준 인터페이스를 갖는 디바이스를 연결하기 위하여 두개의 MII 인터페이스를 가지며, 인터페이스상에 입출력 버퍼를 가지고 있다. 디코더 블록은 IP 패킷 헤더의 특정 필드 (버전, 길이, ToS, Type, 그리고 check sum 등)을 추출하여 미터링 블록으로 전송해 준다. 미터링 블록은 srTCM의 미터링 알고리즘을 수행하고 그 결과를 마킹 기능을 가지고 있는 Tx MII 인터페이스로 넘겨준다. 이 블록은 수신된패킷 정보와 현재의 토큰 버킷값을 기반으로 계산을 수행한 후, 계산 결과에 따라 출력 칼라값과 두개의 토큰 버킷중 하나에 대한 갱신 여부를 결정해 준다. 토큰 갱신 블록은 두개의 토큰 버킷 값 (Bc와 Be)의 현재 값과 초기에 세트된 파라미터 값 (CIR, Unit Time, CBS, 그리고 EBS 등)을 이용하여 주기적으로 계산을 수행한 후 Bc 또는 Be 값의 갱신을 결정한다. 버킷 갱신 블록은 미터링 블록과 토큰갱신 블록으로부터 수신된 Bc와 Bc 값에 대한처리 요구를 수용한다. 만약 이 블록이 미터링 블록 또는 토큰갱신 블록으로부터 Bc 또는 Be 값의 갱신 요구를 받아을 때, 이 블록은 자기가 가지고 있는 현재 값을 수신된 값으로 갱신한다. 그리고 path 블록은 각 블록에게 필요한 제어신호를 제공한다.

그림 8은 srTCM의 미터링 블록의 구성도를 보여준다. 이 블록은 color-blind 모드와 color-aware 모드를 모두 지원할 수 있다. 이 블록은 미터링 모드와 칼라 값을 수신한다. color-aware 모드에서, 이 블록은 미터링 모드와 입력 칼라 값 모두를 참조하여 출력 칼라 값을 결정한다. color- blind 모드에서, 이 블록은 미터링 모드 값만 참조하여 출력 칼라 값을 결정한다. 그러므로 srTCM chip은 미터링 모드 값에 따라 DS 네트워크의 엣지 노드 또는

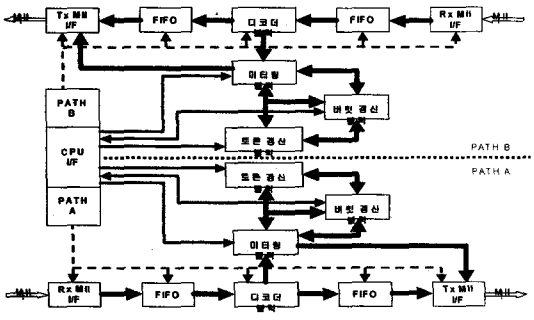


그림 7. srTCM 블록도
Fig. 7. srTCM Block Diagram.

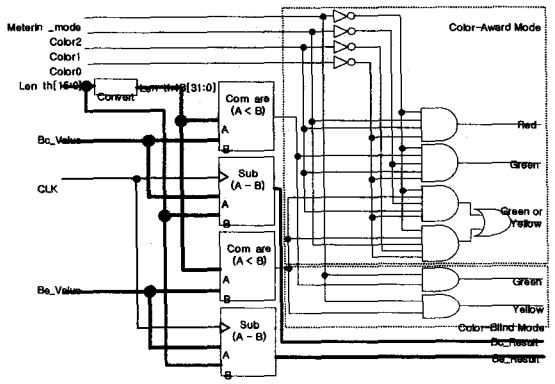
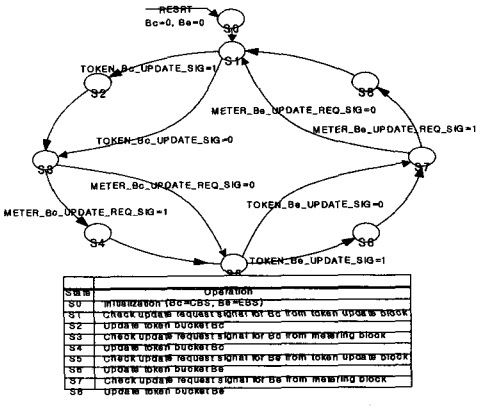


그림 8. 미터링 블록의 구성
Fig. 8. The configuration of Metering block.

코아 노드에서 사용될 수 있다. 이 블록은 각 모드에서 입력 패킷의 길이 값을 현재의 토큰 값과 비교한 후, 그 결과를 마킹 기능을 가진 Tx MII 인터페이스로 전송한다.

그림 9는 버킷 갱신 블록의 동작 흐름도를 보여준다. 이 동작은 총 9개의 상태로 구성된다. 이 블록은 토큰 갱신 블록 또는 미터링 블록에서 수신한 Bc 또는 Be 값을 수용하여 현재 자신이 가지고 있는 값을 갱신한다. 상태 S0에서, 이 블록은 토큰 버킷 Bc와 Be 값을 그들이 최대 버킷 크기 (CBS와 EBS)로 초기화시켜 준다. 초기화 후 상태 S1에서, 이 블록은 토큰 갱신 블록으로부터 Bc에 대한 갱신 요구 신호가 있는지 점검한다. 만약 그 신호가 있다면, 상태 2에서 이 블록은 자신의 현재 Bc 값을 수신된 Bc 값으로 갱신한다. 상태 1에서 해당 신호가 없으면 상태 S3에서 이 블록은 미터링 블록으로부터 Bc에 대한 갱신 요구 신호가 있는지 점검한다. 만약 그 신호가 있다면, 상태 4에서 이 블록은 자신의 현재 Bc 값을 수신된 Bc 값으로 갱신한다. 버킷 Be에 대하여도 상태 S5~S8에서 같은 방법으로 이루어진다.



State	Operation
S0	initialization (Bc=CBS, Be=EBS)
S1	Check update request signal for Bc from token update block
S2	Update token bucket Bc
S3	Check update request signal for Be from metering block
S4	Update token bucket Be
S5	Check update request signal for Be from token update block
S6	Update token bucket Be
S7	Check update request signal for Be from metering block
S8	Update token bucket Be

그림 9. 토근 버킷 갱신흐름도
Fig. 9. Updating Flow of Token Bucket.

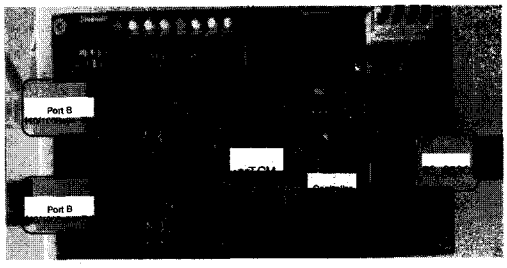


그림 10. srTCM 보드 구성도
Fig. 10. srTCM Board Configuration.

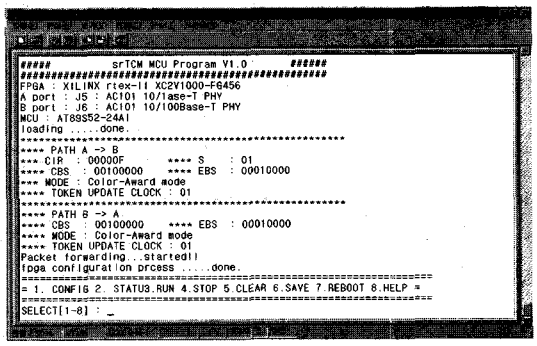


그림 11. srTCM 보드의 상태 화면
Fig. 11. Status Display of srTCM Board.

그림 10은 srTCM chip를 검증하기 위하여 제작된 보드를 보여주고 있다. 이 보드는 두개의 10/100Base-Tx 이더넷 인터페이스, 하나의 RS-232C 포트, 그리고 프로세서로 구성되어 있다. 이 보드를 통하여 포트 A에서 B로 또는 역으로 경로를 설정하여 srTCM 기능을 검증하였다.

그림 11은 하이퍼 터미널을 통하여 세팅된 srTCM의 파라미터 값을 확인할 수 있고, 필요시 해당 값을 수정할 수 있는 화면을 보여주고 있다. 이것은 두개의 토큰 버킷 Bc와 Be에 대한 파라미터 값 (CIR, Unit Time, CBS, 그리고 EBS 등), 미터링 모드 (color-blind 모드 또는 color-aware 모드), 그리고 토큰을 갱신할 때 사용할 클럭 값을 보여준다. 그리고 위의 값들을 메뉴를 통하여 수시로 변경하고 확인할 수 있으며, srTCM이 동작하고 있는 동안 여러 가지 레지스터 값을 확인 할 수 있다. 이들 레지스터들은 정상적인 송수신 패킷 수, 에러가 있는 송수신 패킷 수, IPv4 패킷이 아닌 패킷의 송수신 수, 태그가 된 IPv4 패킷의 송수신 수, 태그가 되지 않은 IPv4 패킷의 송수신 수, 토클 버킷의 토큰 값, 그리고 미터링 결과에 따른 토큰 버킷의 토큰 값을 보여준다.

그림 12는 srTCM 보드를 시험하기 위한 구성도를 보여준다. 실험을 위하여 프로토콜 분석기를 이용하였으며, PC를 통하여 트래픽 발생, 트래픽 분석, 그리고

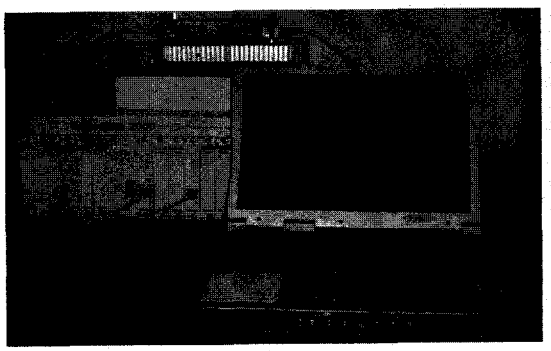


그림 12. srTCM 시험 구성도
Fig. 12. Test Configuration of srTCM.

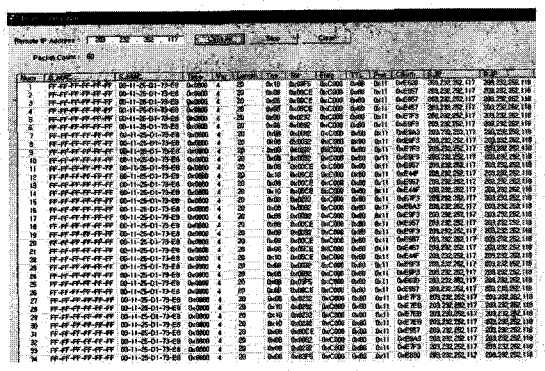


그림 13. Color-blind 모드에서 srTCM 동작
Fig. 13. srTCM Operation in Color-Blind Mode.

srTCM의 파라미터 세팅 및 확인을 수행할 수 있다. Color-award 모드를 선택한 경우에, ToS 값을 설정할 수 있다. DS 네트워크에서 마커 기능을 수행하는 srTCM은 패킷 분류기로부터 패킷을 넘겨 받기 때문에 패킷 클래스를 나타내는 DSCP 값의 상위 3비트는 관여하지 않고, 칼라 값을 나타내는 하위 3비트를 관여한다. 그래서 8비트로된 ToS 바이트의 상위 3비트와 사용하지 않는 하위 2비트는 0으로 간주하였다. 따라서 칼라 값은 <표 1>에 의하여 Green은 0x08, yellow은 0x10, 그리고 red는 0x18를 갖도록 하였다.

이 실험을 위하여 IPv4 패킷을 랜덤하게 발생하였으며, srTCM 파라미터는 다음과 같이 설정하였다: CIR=16Mbps, S (Unit Time)=1, CBS=2300bytes, EBS=4096bytes, 그리고 clock rate=10ns.

그림 13은 srTCM이 color-blind 모드로 동작하는 경우 트래픽 분석기의 출력을 보여주고 있다. 이 분석기는 소스/목적지 MAC 주소, Type 값 (IP 패킷을 나타냄), IP 패킷 버전(IPv4), 패킷 헤더길이, ToS 값, 총 패킷 길이 등을 보여 준다. srTCM은 입력 패킷의 길이를 현재 토큰 값과 비교하여 그 결과를 ToS 필드에Green, Yellow, 그리고 Red로 마킹해 준다. 트래픽 분석기는 srTCM이 패킷에 마킹한 칼라 값을 보여주고 있다.

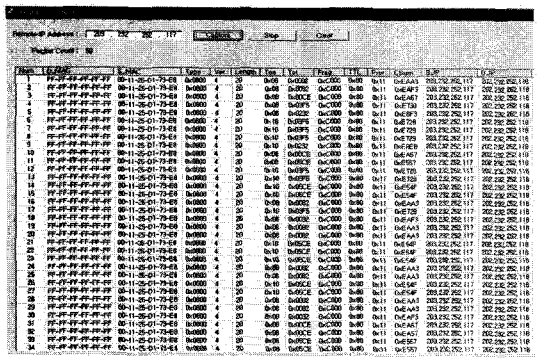


그림 14. Color-aware 모드에서 srTCM 동작 (input color=green)

Fig. 14. srTCM Operation in Color-Aware Mode (input color=green).

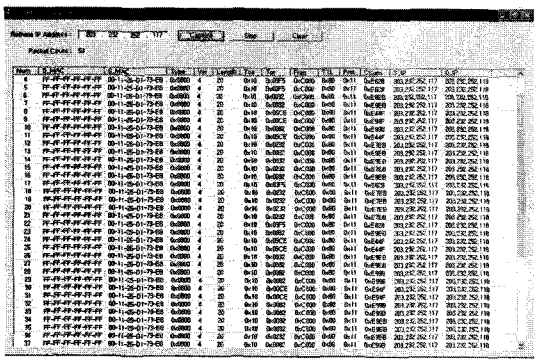


그림 15. Color-aware 모드에서 srTCM 동작 (input color=yellow)

Fig. 15. srTCM Operation in Color-Aware Mode (input color=yellow).

그림 14는 srTCM이 color-aware 모드로 동작할 때, 입력 트래픽의 칼라값이 Green (0x08)인 경우 트래픽 분석기이 출력을 보여주고 있다. 이때 ToS 값이 Green 또는 Yellow 값을 갖는 것을 볼 수 있다. 따라서 이 패킷을 수신한 노드는 Yellow 패킷인 경우, 해당 패킷을 best effort 서비스를 적용하여 포워딩하며, Green 패킷인 경우, 해당 패킷을 낮은 Drop 확률로 포워딩한다.

그림 15는 srTCM이 color-aware 모드로 동작할 때, 입력 트래픽의 칼라값이 yellow (0x10)인 경우 트래픽 분석기이 출력을 보여주고 있다. srTCM chip은 Be에 있는 토큰 값을 입력 패킷의 길이와 비교한다. 패킷 길이가 Be 값을 초과하지 않은 경우 해당 패킷을 Yellow로 마킹하고, 초과하는 경우 Red 값으로 마킹된다. 따라서 이 패킷을 수신한 노드는 Yellow 패킷인 경우, 해당 패킷을 best effort 서비스를 적용하여 포워딩하며, Red 패킷인 경우, 해당 패킷을 Drop할 수 있다.

그림 16은 color-aware 모드에서 Red 패킷을 발생시키는 것을 보여준다. ToS 값을 0x18로 하여 Red 패킷

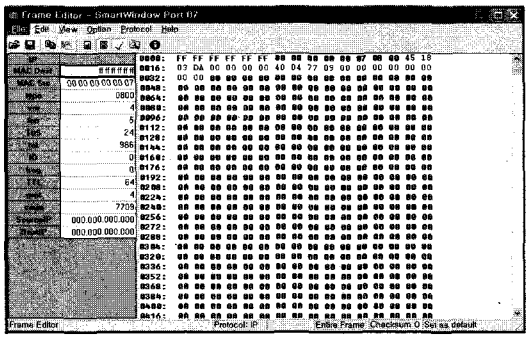


그림 16. Color-award 모드에서 Red 패킷 입력

Fig. 16. Red Packet Input in Color-Aware Mode.

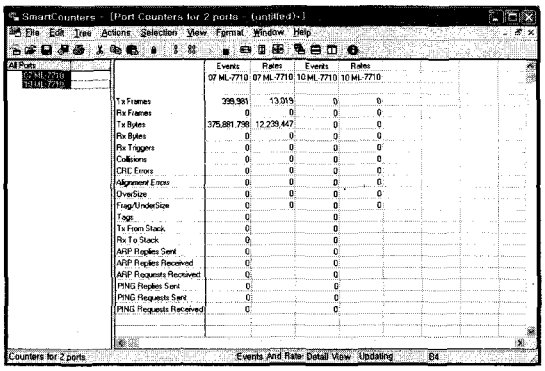


그림 17. Color-award 모드에서 Red 패킷 수신

Fig. 17. Red Packet Receive in Color-Aware Mode.

을 나타낸다. 이 경우 Red 패킷의 처리는 srTCM 개발자의 의도에 의하여 처리하도록 권고하고 있다. 따라서 본 논문에서는 srTCM이 Red 패킷 입력시 폐기하도록 구성하였다.

그림 17은 color-aware 모드에서 Red 패킷으로 세팅된 패킷을 수신한 경우 srTCM의 동작을 나타낸다. 본 논문에서는 이 패킷을 삭제하는 것으로 구성하였기 때문에 그림에서 수신된 패킷이 하나도 없는 것을 보여준다.

srTCM에서 토큰과 관련한 파라미터들을 (CIR, S, CBS, EBS, 그리고 clock rate) 초기화 함으로써, 미터링 기능을 구성하였다. 이때 CBS 또는 EBS 값은 패킷 스트림중 가장 큰 IP 패킷의 길이와 같던지 크게 설정하도록 권고하고 있다. 버킷 갱신 기능은 두 가지 경우로 수행된다. 하나는 토큰 갱신 기능이 새로운 토큰(credit)를 수신하였을 때, 토큰 값을 갱신한다. 다른 하나는 새로운 패킷이 srTCM에 도착하여 미터링 동작이 수행된 경우 미터링 블록에서 토큰 갱신 요구가 있을 때이다. 미터링 기능은 패킷의 길이를 현재의 토큰 값과 비교하여 해당 패킷이 in-porfile인지 out-profile인지 결정해서, 서로 다른 컨디셔닝 동작이 수행될 수 있

도록 그 결과를 다른 트래픽 컨디셔닝 요소로 보낸다. 이 기능은 마킹 기능을 가진Tx MII 인터페이스에서 수행된다. 이것은 미터링 기능의 측정 결과를 패킷 헤더의 DS 필드에 반영하는 것으로, 해당 패킷에 특정DS behavior aggregate가 할당된다.

V. 결 론

본 논문에서는 RFC2697에서 제안된 srTCM을 VHDL과 FPGA 기술을 이용하여 설계하여 구현하였다. 이를 위하여 srTCM의 구조와 동작 알고리즘을 분석하여 기능 블록을 정의하였다. 그리고 srTCM을 하나의 chip으로 구현하고 이를 검증하기 위한 보드를 설계 제작하였다. 이 논문에서는 두개의 토큰 버킷의 파라미터들을 설정하고 확인 할수 있도록 구성하였다. 이 chp은 패킷 헤더의 DS 필드 (ToS 필드)를 특정 DSCP 값으로 할당 (또는 재할당)할 수 있어, 패킷에 특정 DS behavior aggregate을 할당할 수 있다. 그리고 이 chip은 color-blind 모드 또는 color-aware 모드로 동작할 수 있기 때문에, DS 네트워크에서 에지 또는 코아노드에 사용될 수 있다.

추후 이 chip에 사용된 기술을 이용하여 RFC2963에서 제안된 rate adaptive shaper (srRAS)를 구현할 예정이다. 이 shaper는 FIFO, srTCM, 그리고 가상 메모리로 구성되며, AF PHB를 제공하는 DS 네트워크의 ingress에 사용될수 있다.

참 고 문 헌

[1] Stinivas Vegesna, "IP Quality of Service", Cisco Press. 2001.
[2] Brian E. Carpenter and Kathleen Nichols, "Differentiated Services in the Internet," Proceeding of the IEEE, VOL.90, NO.9, September 2002.
[3] Chuck Semeria, John W. Stewart III, "Supporting Differentiated Service Classes in Large IP Networks," White Paper, Juniper Network, Inc. 2001.
[4] Heinanen J. and R. Guerin, "A Single Rate Three Color Marker", RFC2697, September 1999.
[5] K. nicholas, S. Blake, F. Barker and D. Black, "Definition of the Differentiated Service Field (DS Field) in the IPv4 and IPv6 Headers", RFC2474, Dec. 1998.
[6] S. D'Antonio et al., "An Architecture for Differentiated Service", RFC2475, Dec. 1998.
[7] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, "Assured Forwarding PHB Group", RFC2597, June 1999.
[8] Douglas J. Smith "HDL Chip Design A Practical Guide for Designing, Synthesizing and Simulating ASICs and FPGAs using VHDL or Verilog," Doone Publications, 1996.
[9] Charles H. Roth, Jr. "Digital Systems Design Using VHDL", PWS Pbblishing Company, 1998.

저 자 소 개

박 천 관(정회원)
2004년 10월 전자공학회논문지
제 41권 TC 제 10호